Optimization Principles for Hardware/Software Co-design with Applications in Molecular Dynamics

Principle Investigator:
Stephan Eidenbenz,
CCS-3;
eidenben@lanl.gov

Co-PI:
Kei Davis, CCS-7;
kei@lanl.gov

Co-PI:
Arthur Voter, T-1;
afv@lanl.gov

***Key Personnel*:**
Hristo Djidjev
Sue Mniszewski
Danny Perez
Nandakishore Santhi
Sunil Thulasidasan

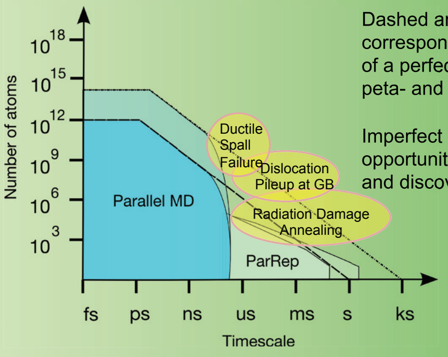
This project is supported by LANL LDRD

Abstract

We propose to develop a framework for *hardware-software co-design as a formally posed optimization problem.* While the optimization framework will be applicable to multiple problem domains, for the target application we use *molecular dynamics (MD)*, an exemplar for the need for computational scaling, and archetypical of the obstacles thereof. We view codesign as search and selection from a vast space of hardware and software designs that map to performance metrics. The objective function that we aim to optimize has as main components *run time (or computational rate), problem size, simulated time duration, energy use, and hardware cost.*

Science Strategy

From our chosen applications domain perspective, we will develop atomistic simulation tools that will enable the study of processes such as *ductile spall failure* under shock conditions and the *evolution of radiation damage*. Achieving this requires a two order-of-magnitude increase in simulated time over current state-of-the art petascale computing, and, more important, cannot be realized without this co-design approach, as direct implementation on an exascale platform cannot achieve this. These two applications are of great intrinsic interest, addressing issues of the response of materials in extreme conditions and enabling the design of more effective and safe fission power plants.

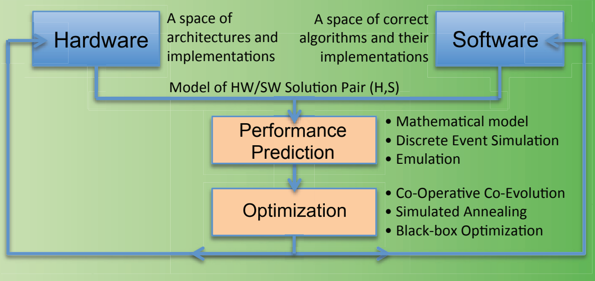


Dashed and dash-dotted lines correspond to the theoretical limit of a perfectly-scaling MD code at peta- and exa-scale, respectively.

Imperfect scaling shows an opportunity for both Co-Design and discovery at current scale.

Co-Design Strategy

Our semi-formal codesign optimization framework relies on (i) efficient enumeration methods for finding feasible hardware architectures and software designs; (ii) a multi-scale approach to performance prediction modeling, where we use cycle-accurate virtual machine emulation, discrete event simulation, graph mapping, and constraint programming as different prediction methodologies; and, (iii) optimization methods with fast identification of new hardware-software pairs to be tested with the more detailed performance prediction methods. We define hardware and software designs in a hierarchical fashion. Enumeration of software designs will initially be via correctness-preserving transformations. The enumeration of hardware architectures is done implicitly through parameter spaces and optimization search methods as well as (human, possibly machine-aided) design feasibility checking. Our performance prediction methods model at different levels of detail, thus covering the trade-off space of accuracy versus scalability in both time and size; performance prediction techniques we will employ include constraint mathematical programming, discrete event simulation, and virtual machine emulation.



Hardware

A space of architectures and implementations

A space of correct algorithms and their implementations

Software

Model of HW/SW Solution Pair (H,S)

Performance Prediction

- Mathematical model
- Discrete Event Simulation
- Emulation

Optimization

- Co-Operative Co-Evolution
- Simulated Annealing
- Black-box Optimization

Optimization heuristics are necessary for the more detailed prediction methods of virtual machine emulation and discrete event simulation. Our framework allows for the adaptation of standard meta-heuristics such as simulated annealing, taboo search, gradient search, and genetic algorithms. The more coarse-grained prediction methods of graph mapping and constraint programming will optimize through graph algorithms and mathematical programming techniques.

LA-UR 11-06271